IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 21, 2021, accepted April 2, 2021, date of publication April 13, 2021, date of current version April 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073057

WINDS: A Wavelet-Based Intrusion Detection System for Controller Area Network (CAN)

MEHMET BOZDAL, MOHAMMAD SAMIE, AND IAN K. JENNIONS
IVHM Centre, Cranfield University, Bedford MK43 0AL, U.K.

Corresponding author: Mehmet Bozdal (mehmet.bozdal@cranfield.ac.uk)

This work was supported by the Republic of Turkey Ministry of National Education.

ABSTRACT Vehicles are equipped with Electronic Control Units (ECUs) to increase their overall system functionality and connectivity. However, the rising connectivity exposes a defenseless internal Controller Area Network (CAN) to cyberattacks. An Intrusion Detection System (IDS) is a supervisory module, proposed for identifying CAN network malicious messages, without modifying legacy ECUs and causing high traffic overhead. The traditional IDS approaches rely on time and frequency thresholding, leading to high false alarm rates, whereas state-of-the-art solutions may suffer from vehicle dependency. This paper presents a wavelet-based approach to locating the behavior change in the CAN traffic by analyzing the CAN network's transmission pattern. The proposed Wavelet-based Intrusion Detection System (WINDS) is tested on various attack scenarios, using real vehicle traffic from two independent research centers, while being expanded toward more comprehensive attack scenarios using synthetic attacks. The technique is evaluated and compared against the state-of-the-art solutions and the baseline frequency method. Experimental results show that WINDS offers a vehicle-independent solution applicable for various vehicles through a unique approach while generating low false alarms.

INDEX TERMS Controller area network, intrusion detection, in-vehicle network, wavelet analysis.

I. INTRODUCTION

Vehicles are getting more connected and autonomous year by year due to communication between Electronic Control Units (ECUs), which control one or more vehicle functions such as engine control, telematics control, and airbag deployment. There are various established in-vehicle communication standards, such as Controller Area Network (CAN), FlexRay, Local Interconnect Network (LIN), and Media Oriented Systems Transport (MOST) [1]. Among these, CAN is the most widely used in-vehicle communication protocol [2] because of its recognized advantages in robustness, suitability for real-time networks, easy maintenance, and low-cost implementation. However, it does not have any intrinsic security features for protecting it against cyberattacks. The vulnerabilities of CAN were presented for the first time by Hoppe *et al.* [3], [4]; since then, researchers have demonstrated a variety of physical and remote access attacks [5]–[7]. The increasing number of attacks shows that the protocol is defenseless to cyberattacks. Overcoming such security shortcomings relies on developing efficient prevention mechanisms, which with the current state of the art fall into four categories: encryption, authentication, network segmentation, and Intrusion Detection Systems (IDSs).

Lack of encryption and authentication is the leading root cause of CAN vulnerability. Although cryptographic techniques are the direct solution, implementation of such algorithms is not feasible for CAN in automotive applications because of limited resources (bandwidth and computational power), the need for long service life, and time constraints. Researchers [8] have shown that current cryptographic methods are not suitable for commercial vehicles due to significant overhead or backward incompatibility.

Network segmentation, which limits access to the critical ECUs by separating them from the user-accessible network via a gateway, is not secure enough to stop adversaries. There are successful attacks that pass the gateway ECU and intrude to the in-vehicle network [9].

IDS can provide adaptable protection by monitoring the CAN network and labeling the malicious messages without modifying the legacy ECUs. Different IDS approaches are applied to mitigate the security problem of the CAN network. Some of these solutions are developed based on promising machine learning techniques like Hierarchical Temporal

The associate editor coordinating the review of this manuscript and approving it for publication was Pedro R. M. Inácio.

Memory (HTM) [10], Generative Adversarial Nets (GAN) [11], Long Short-term Memory (LSTM) [12], and other deep neural networks [13], [14]. However, such methods suffer from high computational power. Additionally, these methods are heavily vehicle dependent and require specific training for different vehicle makes and models. Similarly, entropy-based IDSs [15]–[17] need training to detect anomalies. They are also highly vulnerable to attacks that do not change the entropy, for instance, replay attacks. Other researchers have applied specification-based IDS solutions [18], [19] by creating rules based on the protocol specification. However, these solutions are protocol depended and can fail if an attacker mimics the message flow sequence. Although IDS is a promising path to address CAN vulnerability, by labeling malicious messages despite the limited resources, available IDSs have major weaknesses [20] such as high false-positive rate, vulnerability to certain attack types, and vehicle dependency. Many IDS solutions do not even consider the detection time, which has an enormous impact on real-time systems.

The proposed vision to address this problem is to explore techniques that speed up attack detection time and reduce the IDS' decision-maker unit's dependency on prior knowledge, intending to reduce the rate of false alarms, which ultimately increases attack detection accuracy. In this regard, the paper contributes to identifying malicious messages by analyzing network traffic behavior rather than its frequency, using wavelet analysis. The main contributions of the paper are:

- A novel, fast detection wavelet-based IDS for in-vehicle networks
- A vehicle independent IDS approach for attack detection without prior knowledge
- Evaluation of the proposed method on real vehicle data and comparison with state-of-the-art methods

The remainder of this paper is divided into the following five sections. Section II provides background information on the CAN network, wavelet analysis, intrusion detection systems, along with related works and CAN bus attacks. The WINDS algorithm and experimental setup are presented in Section III and Section IV, respectively. Section V shows the results along with the discussion. Finally, Section VI presents future directions, and Section VII concludes the paper.

II. BACKGROUND

A. CAN PROTOCOL

The CAN bus is a multi-master broadcast communication interface designed for in-vehicle communication. The traditional CAN standard's speed is limited to 1 Mbps with 8-byte payload transmission in a frame, while the newer version CAN FD (Flexible Data-rate) reaches 64-byte payload transmission [21]. The protocol has message-based communication via frames, consisting of a message identifier field, data field, control bits, and Cyclic Redundancy Checksum (CRC) [21]. Every node listens to each broadcasted frame and processes the relevant ones based on the message identifier field. This field is also used for the arbitration mechanism allowing the higher-priority node to transmit without collision when two or more nodes transmit simultaneously. The lower message identifier has a higher priority.

The protocol utilizes differential signaling lines, known as CAN high and CAN low, on the twisted wire. Hence the data is presented by voltage difference; the network is resilient to electrical noises. The signals are logically presented by the recessive ''one'' and dominant ''zero'' bits. The dominant bit can overwrite the recessive one, which means the bus signal is dominant if nodes transmit the complement signals simultaneously. The standard also protects the bus from an unhealthy ECU by Error Confinement Mechanism (ECM) [21]. ECM includes two error counters that their values increase by the relevant transmitting or receiving errors, respectively. If any of the counter values exceed the limit, the node goes to the buss-off state and will not transmit data.

B. WAVELET TRANSFORM

Wavelet analysis provides a frequency analysis of the signal and gives information about breakpoints, trends, and self-similarity. It is used in various fields, including information security, oceanography, medicine, and finance. Unlike the Fourier Transform, it gives frequency analysis on the time domain. Continuous Wavelet Transform (CWT) converts signal $f(t)$ into wavelet coefficients $F(a, b)$ which is a function of scale a and position b as defined below:

$$F(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi^* \left(\frac{t-b}{a} \right) dt \quad (1) \quad [22]$$

where ψ is called mother wavelet, which is any function that satisfies:

$$\int_{-\infty}^{\infty} \psi(t)\, dt = 0 \quad (2) \quad [22]$$

$$\int_{-\infty}^{\infty} \psi^2(t)\, dt = 1 \quad (3) \quad [22]$$

Scaling means compressing or stretching the mother wavelet. While the compressed wavelet provides rapidly changing high-frequency information, the stretched one gives details of slow changes. The scaling feature offers local and global details of the signal. Unlike Discrete Wavelet Transform (DWT), which has a decreasing number of coefficients with increasing scaling factor, CWT has the same number of coefficients at each scale. This redundancy (i.e. has the exact time resolution as the original data) of CWT provides a more accurate time-frequency spectrum.

C. INTRUSION DETECTION AND RELATED WORK

An IDS can be categorized as signature-based and anomaly-based. Signature-based IDS, as shown in Figure 1.a, has an attack (signature) database and works similar to anti-virus software. If an attack from the database occurs, it can identify the attack. On the other side, anomaly-based IDS, as shown in Figure 1.b, characterizes the system's behavior and compares it with baseline and alerts if the deviation

TABLE 1. Summary of recent intrusion detection systems for CAN bus.

Ref.	Parameter	Algorithm / Method	Advantages	Downsides
[23]	Electrical signal	Support vector machine and boosted decision tree	Robust to some attack types, differentiate between an error and an attack	High cost and vulnerability to environmental conditions
[24]	Electrical signal	Multilayer perceptron	Robust detection of malicious nodes	High cost and vulnerability to environmental conditions
[25]	Time intervals	Recursive least squares and Cumulative Sum (CUSUM)	Identification of attacked ECU	Works only for periodic signals, susceptible to environmental conditions
[11]	A pattern of CAN ID	Generative adversarial nets	Robust to attacker manipulation	Heavy resource usage, vehicle dependency
[14]	Traffic pattern	Deep convolutional neural network	Better performance than other machine learning methods	Heavy resource usage
[32]	Timing analysis	Specification-based	Low computational requirement	Defining the specifications
[33]	Remote frame Timing	Offset ratio	Efficient and straightforward algorithm with low-cost hardware	Increased traffic
[34]	Period and payload	Bloom filtering	Low memory usage	High computational power
[35]	Time intervals	Z-score and ARIMA	Minimal training	High time-to-detect

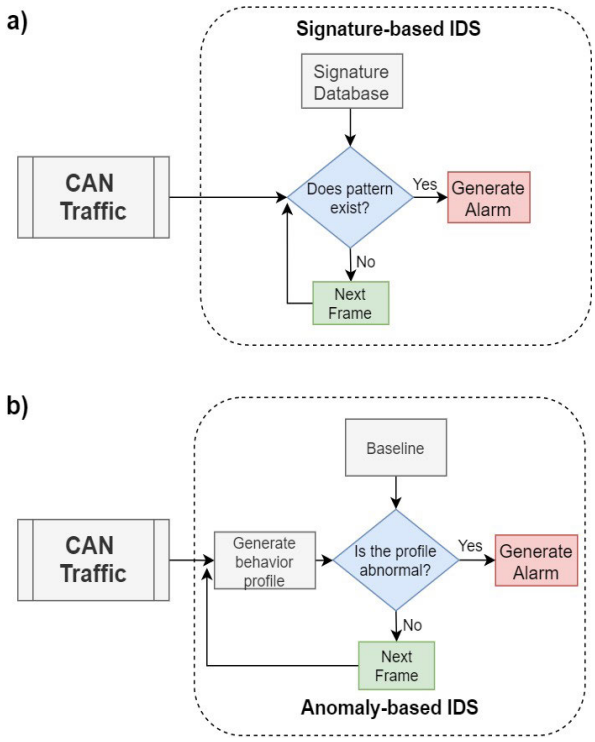

FIGURE 1. (a) Flowchart of signature-based and (b) anomaly-based intrusion detection systems.

from the baseline exceeds a certain threshold. Although signature-based IDS is quite successful for known attacks, it cannot detect unknown attacks and requires a regular update of the database. Hence, it is impossible to know that all the attacks and regular updates can be a hassle; anomaly-based IDS solutions have advantages over signature-based ones.

The current IDS solutions, as summarized in Table 1, use various parameters to analyze the CAN network. Research in [23]–[25] take advantage of the physical characteristic of the network. Thanks to random manufacturing variations, cabling, and aging, each transceiver has a slightly different signature on the signal even though they transmit the same data. Analyzing these signatures gives the means to identify authentic messages. Although these methods are highly reliable in a controlled environment, their performance changes significantly based on environmental change like temperature. They are also vulnerable to detect malicious messages from the software layer, as explained in [5].

Müter *et al.* [26] identify eight anomaly detection sensors that provide the essential input to structure an in-vehicle network. These are frequency, formality, location, range, correlation, protocol, plausibility, and consistency. These are not necessarily physical sensors but are signal processing boxes/tools that process the CAN bus's network traffic to observe and monitor changes as for such parameters. Any IDS solutions use one or multiple of these sensors. As many ECUs broadcast CAN frames regularly, frequency is one of the most critical anomaly detection sensors to characterize the automotive network, if not the best. An intrusion into the CAN network will disrupt the regularity of the transmissions and the system's frequency. Although time thresholding is a simple technique to detect attacks, it can generate a high false-positive rate. On the contrary, frequency analysis gives more stable information [27]. Therefore, the CAN network's frequency analysis is a simple but effective IDS solution for resource-constrained vehicles.

There are multiple pieces of research to assess the time interval and frequency of the CAN messages. Some of these use basic statistical analysis [28], [29], but they are

highly vehicle-dependent. Machine learning algorithms like One-Class Support Vector Machine (OCSVM) [30], Gaussian mixture model [31] also proposed to detect anomalies via frame timing analysis; however, they require a comprehensive training data set for each vehicle model. ARIMA and Z-score were proposed [35] to minimize the training phase and vehicle dependency, but a successful result requires a long window size, which will increase the detection time. Lee *et al.* [33] analyzed the response time of the ECUs by sending them remote frames. Their method requires low computational power and is successful in detecting attacks. The downside of the technique is that it increases bus traffic by sending remote frames.

On the other hand, wavelet analysis has outstanding performance, mainly due to its simple procedure, easy computation, and reconstructable decomposition. This motivated researchers from the IT security domain to benefit [36]–[38]. Spicer *et al.* [24] proposed wavelet analysis for CAN bus to complement his noise-content-based multilayer perceptron IDS with frequency analysis. His implementation was limited to the signal level and analyzed the electrical characteristic to identify different signatures. By fingerprinting ECUs, it is possible to identify the sender ECU; therefore, the work can also be regarded as an authentication method. The work presented in this paper moves beyond Spicer's research and intends to develop the entire IDS based on wavelet analysis. The WINDS is applied to message level and analyses behavior of message frequency, facilitating low-latency frequency analysis for the CAN network without increasing the network traffic and training data requirement.

D. CAN BUS ATTACKS

There are multiple attack types applied to the in-vehicle networks. Suppose an attacker has direct access to the CAN bus. In that case, she/he can read and write to the CAN network, proceed with overwriting and invalidating legitimate messages, and further disable a CAN node. In this circumstance, the following attacks are possible to implement:

1) DENIAL OF SERVICE (DOS)

An attacker can send high-priority CAN messages and holds the bus in busy condition; therefore, other low-priority nodes cannot access the network. This flooding attack will substantially increase the frequency of the messages. Murvay presented an example of this attack in [7].

2) DROP/SUSPENSION

An attacker can disable a node and suspend the message transmission. It is a subcategory of the DoS attack; hence, it is impossible to get service from the suspended node. Palanca *et al.* took advantage of the CAN protocol's error confinement mechanism and disabled an ECU by transmitting dominant bits over the recessive ones [39].

3) FUZZING

An attacker can send random values without any in-depth knowledge and confuse the network. This attack does not require reverse engineering. Inserting fuzzing messages will increase the frequency of the CAN message like the one in [40].

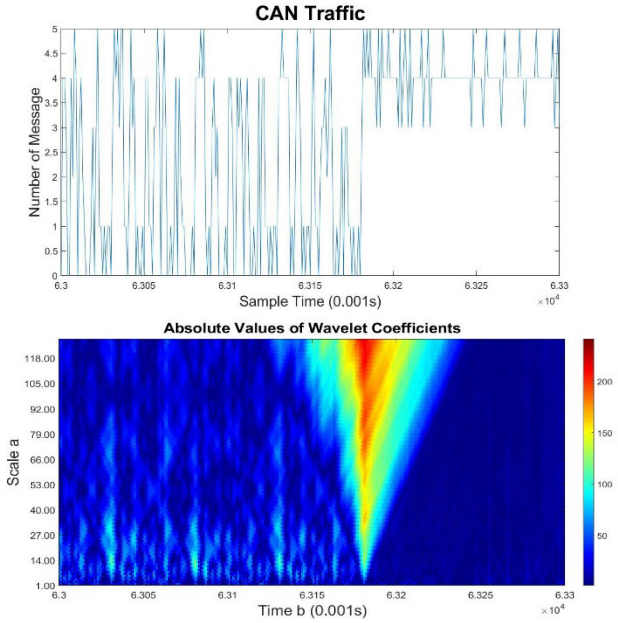

FIGURE 2. Message count of the CAN traffic (top) during a DoS attack and its wavelet analysis (bottom).

4) REPLAY

An attacker can read the CAN messages and send them back to the network later on, such as [41]. Hence, there is no freshness check on the protocol; other nodes will accept the replayed message. The authentic node sends messages at a particular frequency, but the attacker overwrites the original command by replaying CAN messages at a higher frequency.

All these attacks disrupt the system's behavior and result in frequency deviation. The attack detection mechanism of the WINDS and other frequency/time-based IDS solutions rely on identifying these variations.

III. SECURING THE CAN NETWORK VIA WAVELET ANALYSIS

The frequency profile contains essential information about the authenticity of CAN messages when obtained by the Continuous Wavelet Transform (CWT). CWT is a powerful tool for the precise localization of frequency components on the time axis, useful for identifying irregularities in the CAN network's traffic pattern. In order to find the signal's behavior change, WINDS benefits from CWT for dividing the network pattern, which is a continuous time-series signal, into different scale components. Then the analysis is further carried out on the scale domain. Figure 2 visualizes the CAN traffic and its wavelet representation. The figure depicts a set of large CWT coefficients located vertically around $t = 6.318(s)$ where the change (attack) occurs in the signal. The area of large coefficient values, which is called the cone of influence, spreads with rising scale but still centered

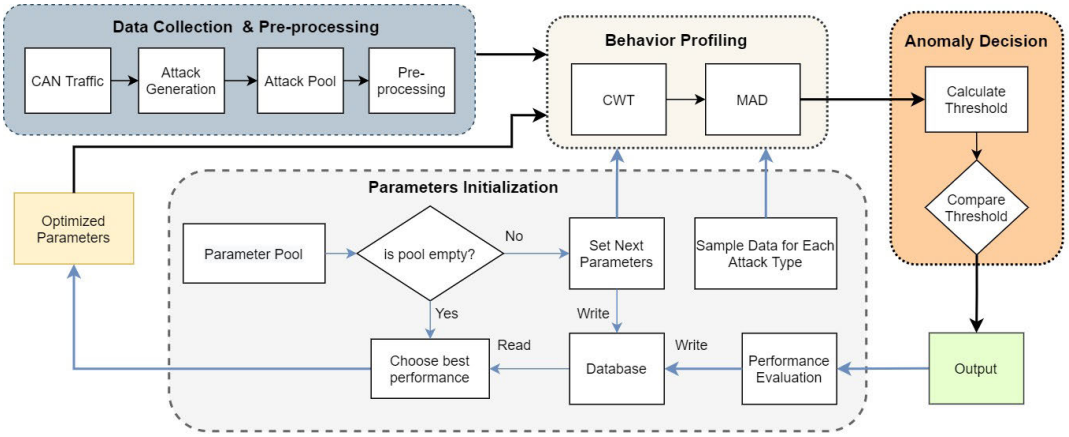

FIGURE 3. The flowchart of wavelet-based intrusion detection system for in-vehicle communication.

at t = 6.318s. It presents us which CWT coefficients are affected by the signal at that point. Therefore, the proposed WINDS algorithm can detect both long-time and sudden short-time duration attacks by analyzing scales.

The WINDS algorithm can be split into four stages, as shown in Figure 3: data collection and preprocessing, behavior profiling with CWT, anomaly decision, and parameter initialization.

A. DATA COLLECTION AND PREPROCESSING

The first stage is to monitor the CAN traffic under various no-attack and attack scenarios. This is a time-consuming data creation task and requires multiple resources and tools. Several research centers lead such experiments and data collection steps, providing researchers with valuable datasets. Although open-access datasets might be limited to specific cases, they can be well extended to comprehensive data by considering various attack models and understanding the CAN bus system's technical details and the vehicle's performance. This usually turns in populating the initial experimentally collected dataset with several synthetic attacks that mimic attacks presented in Section II.D.

The preprocessing step starts with windowing the dataset, proceeded with a feature extraction step, which is usually conducted by the signal-processing tool. Assuming a widowed data as $w(t)$, it is a collection of messages, M, while each has a time interval of sampling time t_s as in (4), representing traces of the message counted over the previous n samples:

$$w(t) = \{M_{t-(n-1)*t_s}, M_{t-(n-2)*t_s}, \ldots, M_t\} \tag{4}$$

The WINDS benefits from message count N_w in the CAN traffic in window w within a specified time interval between t and $t - t_s$, where t_s is the period of the interval. Hence, we specify a message frequency, S_f, with the following equation, applied on i$^{\text{th}}$ window w_i, to account the frequency of message in that window:

$$N_i = S_f\left(M_{w_i}\right) = \sum_{k=1}^{n_{max}} M_k \tag{5}$$

where n_{max} is the maximum number of messages that a window may have, and M_k represents the existence of the k^{th} message within the i^{th} window (w_i) that is one if a message exists; otherwise, it is zero. The window is stretched from the current time to the past, and the analysis is processed frequently. This results in the featured i_{th} window by the frequency conversion S_f, represented by w_i^S, as in the following equation:

$$w_i^S = \left\{N \in Z : \exists N_1, \ldots, N_{n-max} \in w \; with \; N = S_f\left(M_{w_i}\right)\right\} \tag{6}$$

B. BEHAVIOR PROFILING

The second stage is generating the behavior profile from the preprocessed traffic signal using the wavelet transform in (1). It transforms w_i^S to a set of wavelet coefficients $W(a, b)$, which is a two-dimension matrix of $n\,x\,k$ where k is the highest wavelet scale and n is the window size. To decrease the complexity and get meaningful data out of all wavelet scales, Mean Absolute Deviation (MAD) is used, as in (7), where L is the length of scale for the chosen w_i^S after wavelet transformation, j and q denotes a specific component of the scale as an index, and A_i^{MAD} projects the results after applying a MAD function on the scaled component for the i$^{\text{th}}w_i^S$. Therefore, MAD provides the absolute deviations from the mean point and gives information about the wavelet scale changes in each sample. Figure 4 demonstrates an example of MAD transformation from the wavelet coefficients during a replay attack.

$$A_i^{MAD} = \frac{1}{L}\sum_{j=1}^{L}\left|a_j - \frac{1}{L}\sum_{q=1}^{L} a_q\right| \tag{7}$$

C. ANOMALY DECISION

This is the step for interpreting the wavelet coefficients, which leads to change point detection, needed for exploring anomalies' symptoms caused by an attack. The core of anomaly detection is assessing each window to find behavior deviations using a thresholding technique. Donoho and Johnstone [42] proposed a universal

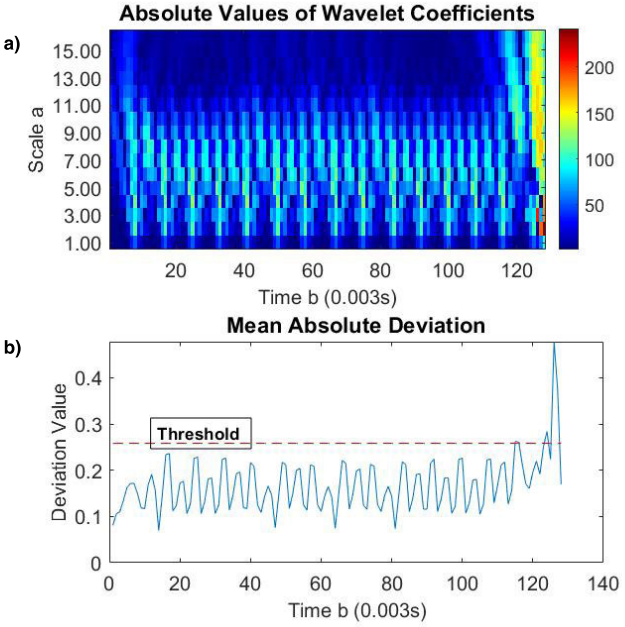

FIGURE 4. The wavelet transform of the windowed signal w (t) for single ID during replay attack (top) and median absolute deviation of W(a, b) (bottom).

threshold λ_u defined as:

$$\lambda_u = \sigma.\sqrt{2\log(N)} \tag{8}$$

where σ is the standard deviation and N is the number of samples. Donoho and Johnstone's threshold technique's advantages are slightly limited to denoising the White Gaussian noise affected signals by finding substantial change. Mozzaquatro *et al.* [43] presented that the universal threshold λ_u should be updated by a constant correction factor ρ to get a better results, (9). The constant correction factor ρ depends to specific applications of interests like anomaly detection for web attacks, boundary conditions, etc.

$$\lambda = \rho.\lambda_u \tag{9}$$

It is known so far that thresholding is the crucial element of an IDS so that low and high thresholds lead to false positives and false negatives results, respectively. WINDS involves an adaptive thresholding technique for increasing the accuracy of decisions and calculating a new threshold for each window by updating the ρ parameter based on each window's MAD value. Finally, the updated λ is applied to the anomaly decision process for denoting the values higher than the threshold as anomalies, and so detection of threat. Figure 4 visualizes the WINDS' thresholding mechanism via an example, demonstrating the results of converting the wavelet coefficients in Figure 4.a into MAD values in Figure 4.b. If any of the MAD values within a window exceeds the threshold, that window is regarded as malicious.

There are conditions in which MAD produces results equal to zero based on the specific attack types. An example is when a flooding attack causes suspension of the messages from lower priority ECUs in the presence of the CAN network's arbitration mechanism. In such cases, the window spans inside the attack duration, which causes all the wavelet coefficients to turn to zero, and as a result, MAD generates zero.

D. PARAMETER INITIALIZATION

The proposed IDS involves a multi-parameter optimization problem that requires extensive time and works to find the best performance for the WINDS. Instead, the effort is put forward to initial the WINDS with the best possible parameters experimentally founded by looking into the performance when feeding WINDS with various datasets. This stage is run only once for gathering the values of the parameters. At first, the ranges of each parameter value are chosen and inserted into the parameter pool. These parameters are wavelet type (Haar and Daubechies), wavelet scale (from 4 to 32), window size (from 32 to 256), window-type (discrete and continuous), sample time (from 0.5 ms to 3 ms), and threshold (from 1.1 × MAD to 2.2 × MAD of the current window). The algorithm is then tested for one attack data for each attack type for all the parameters inserted in the pool. The parameter setting which provides better performance on average is chosen as experimental parameters.

IV. EXPERIMENT AND ANALYSIS

A. DATASET

An essential step for developing an IDS is to test it on comprehensive datasets on vehicles considering various working conditions, attack models, the vehicle made, and driving style. The collection of comprehensive datasets requires running a testing vehicle equipped with measurement instruments on dedicated roads where safety measures are taken. Instead of going through such procedures, we benefit from the open-source datasets from two independent research centers [44], [45], which are well deserved by the research community and have already been cited for many different research pieces. This allows us to develop and compare WINDS with other techniques; further benchmark it for testing the WINDS against different vehicle models and driving styles while avoiding WINDS manipulation for any particular dataset.

Both centers' datasets include the well-recognized attack models, including Daniel of Service (DoS), suspension, replay, spoofing, and fuzzy attacks. The downside of both datasets is only one attack instance is considered for each attack model, which is insufficient for testing IDS' capabilities. To overcome the problem, we studied the CAN traffic for vehicular applications and explored realistic traffics under various driving scenarios and attack models to get a view for intentionally extending the initial datasets to the one that covers more comprehensive attack datasets. This resulted in generating synthetic attacks allowing us to see the capabilities and limitations of the IDS by testing it on different attack scenarios. Applying such a technique to the initial data provides us with:

TABLE 2. Generated synthetic attacks based on automotive CAN bus intrusion dataset v2.

Data Source	Attack Type	# Messages	Malicious Messages	Attack Duration
Vehicle 1	No attack	2690069	-	-
Vehicle 2		386567	-	-
Vehicle 1	Denial of	806999	4000 - 40000	1s to 10 s
Vehicle 2	Service	115971	4000 - 40000	1s to 10 s
Vehicle 1	Suspension	806999	50 – 500 [a]	1s to 10 s
Vehicle 2		115971	40 – 400 [a]	1s to 10 s
Vehicle 1	Replay	806999	8-30	75 ms
Vehicle 2		115971	8-30	66 ms

[a] Number of maliciously deleted messages.

TABLE 3. Car-hacking dataset from real vehicle attack.

Attack Type	# Messages	Malicious Messages	Attack Duration
DoS Attack	3665771	587521	3s to 5s
Gear Spoofing	4443142	597252	3s to 5s
RPM Spoofing	4621702	654897	3s to 5s
Fuzzing Attack	3838860	491847	3s to 5s

- *Synthetic Attacks:* Attacks induced synthetically to real CAN traffic shown in Table 2
- *Real Attacks:* Attacks implemented on a real running vehicle in Table 3

The synthetic attacks are generated from Automotive Controller Area Network (CAN) bus intrusion dataset v2 [44], consisting of synthetic attacks based on real CAN traffic from two commercially available vehicles and will be used for testing WINDS in the following sections. The methodology to generate the artificial attacks to regenerate ten attack data is the same attack methodology as the original attack; however, attack strengths are different. We increased the attack duration gradually, second by second, from 1 to 10 seconds for DoS and suspension attacks. Similarly, the inserted malicious message' frequency rises step by step from the attacked node's base frequency to ten times faster for the replay attack. Simulated attacks presented in Table 2 are more challenging to detect than the original attacks because attack duration is short, and traffic's effect is minimal.

The real attacks are to test the algorithm in a real-world scenario where the targeted vehicle is running. Although synthetic attacks mimic the real ones, they cannot mimic the knock-on effect, which may affect the results. Therefore, WINDS is tested on the Car-hacking dataset [45], has data from an actual vehicle while message injection attacks were performed, as shown in Table 3. DoS attack was implemented by injecting the highest priority CAN messages while fuzzing attack was executed by random CAN ID and payload values. A spoofing attack was implemented by inserting malicious messages on relevant CAN IDs for Gear and RPM.

B. EXPERIMENTAL SETUP

Resulted from the parameter initialization step mentioned in Section III, we set the experimental setup with the specifications given in Table 4. The window $w(t)$ includes 128 samples, consisting of 384 ms network traffic, collected by a sampling time of 3 ms for the ID-based data segments. The setup is tied for short sampling times, ensuring the window would be populated with sufficient active data while avoiding information misses. Short sampling time also results in earlier attack detection, consequently. The threshold is set to 1.8 $\times$ the MAD value of the current window. The Haar wavelet, consisting of shifted and scaled square wave functions, is used as a mother wavelet in the analysis. The Haar function in (10) has the potential for looking at differences of averages, essentially. The initial scale for Haar wavelet in this experiment is 16.

TABLE 4. Experimental setup specifications.

Parameter	Value
Number of samples in w(t)	128
Number of messages in a window, N	Variable*
Network traffic	384 ms
Sampling time	3 ms
Type of wavelet	Haar
Number of scale	16
Constant correlation factor, ρ	∝ to MAD
Threshold	1.8 x MAD

* Number of the messages depends on the traffic and ID of ECU.

TABLE 5. Confusion matrix for IDS decision.

		Actual Situation	
		Attack	No Attack
IDS Decision	Attack	TP	FP
	No Attack	FN	TN

$$\psi^{(H)}(t) = \begin{cases} 1, & 0 \le t < \frac{1}{2}; \\ -1, & \frac{1}{2} \le t < 1; \\ 0, & otherwise. \end{cases} \quad (10)\ [46]$$

C. PERFORMANCE EVALUATION METRICS

The evaluation is carried out by assessing each window and labeling them based on the confusion matrix in Table 5. There are four possible outcomes of this decision:

True Positive (TP): A malicious message detected correctly.

False Positive (FP): An authentic message is detected as a malicious message.

True Negative (TN): An authentic message is detected correctly.

False Negative (FN): A malicious message is detected as an authentic message.

As accuracy does not always represent the real success of the IDS (i.e., when data is not symmetric), True Positive Rate (sensitivity), False Positive Rate (FPR), and Precision are also

TABLE 6. The performance of WINDS for the synthetically generated data.

Data Source	Attack Type	Accuracy	Sensitivity	FPR	Precision	MTTD (s)
Vehicle 1	No attack	0.9997	-	0.0003	-	-
Vehicle 2		0.9999	-	0.0001	-	-
Vehicle 1	Denial of	0.9995	0.9982	0.0004	0.9920	0.003
Vehicle 2	Service	1.0000	0.9979	0.0004	0.9985	0.006
Vehicle 1	Suspension	0.9999	0.9980	0.0001	0.9879	0.003
Vehicle 2		0.9997	0.9981	0.0003	0.9993	0.006
Vehicle 1	Replay	0.9998	0.9893	0.0001	0.9986	0.003
Vehicle 2		0.9997	0.9974	0.0003	0.9863	0.003

Mean value of results for the ten datasets.

calculated using the following equations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{12}$$

$$\text{FPR} = \frac{FP}{TN + FP} \tag{13}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{14}$$

In which sensitivity presents the IDS probability to detect an attack, whereas FPR is the probability of labeling an authentic message as an attack. Precision, positive predictive value, shows how accurately the algorithm label the malicious messages.

Additionally, the Time-To-Detection (TTD), the time difference between the attack start and the time that the algorithm detects an attack [47], is also considered as a performance metric and calculated by the following formula:

$$t_{TTD} = t_D - t_s \tag{15}$$

where t_{TTD} is TTD, t_s is the time attack started, and t_d is the time the algorithm detected the attack. Mean Time-To-Detect (MTTD), the average time between the attacks start and the attacks' detection, is calculated by averaging the TTD durations.

V. RESULTS AND DISCUSSIONS

A. RESULTS

The WINDS was tested on two groups of the datasets collected from three commercial vehicles. The first experiment evaluates WINDS capabilities on a broad range of synthetic attacks with varying attack strength. The second experiment assesses the performance of WINDS on a real vehicle attack dataset and compares it with existing solutions.

1) SYNTHETIC ATTACKS

The synthetic attacks allow us to safely implement various attack scenarios, facilitating the observation of the IDS' performance and limitation on different attacks by tuning the attack strength and duration. Using such techniques, we tested WINDS on various attack scenarios, including DoS, suspension, and replay attacks, along with an attack-free dataset.

The experiment is constructed on splitting the entire network data into segments based on the ID numbers, then proceed with analyzing each ID-based data separately to get satisfactory results, shown in Table 6.

The attack-free dataset gives information about how IDS will perform and react in the normal traffic mode by looking into evaluation metrics such as FPR rate. It is essential to keep FPR low; otherwise, higher rates generate many false alarms, which drivers may ignore. Moreover, a higher FPR rate hardens the tasks of the security team. The WINDS' FPR rate is kept for less than 0.0004.

Denial of Service (DoS) attack by flooding high-priority messages can significantly affect network behavior. Although the attack is implemented by sending messages with the highest priority (CAN ID '000'), it can be detected by monitoring any ID in the network. The WINDS algorithm successfully detected DoS attacks, with an average, attack detection rate of 99.82% and 99.79% for Vehicle 1 and Vehicle 2, respectively. The detection rate can reach as high as 99.94% for more prolonged attack durations. The attacks were also swiftly detected in less than 6 ms.

Suspension attack has similar results to DoS attacks as shown in Figure 5. It could be anticipated the same because the arbitration scheme does not allow lower priority nodes to transmit when the DoS attack is implemented. Therefore, suspension attack mimics the DoS attack. The average sensitivity values for Vehicle 1 and Vehicle 2 were 99.8% and 99.81%, consecutively.

Replay attacks were implemented for a short duration of time (75 ms and 66 ms) with low message insertion rates (from 8 to 30 frames). The results shown in Figure 6 depict that the WINDS algorithm can respond in milliseconds and successfully detect over 96% of the attacks with almost zero false-positive rate. The observation is that the algorithm's sensitivity rises with the increased rate of malicious messages while the TTD decreases.

The experimental results show that the sensitivity of the WINDS is correlated with the attack strength. In general, the sensitivity increases for the longer duration and more frequent attacks, as seen in Figure 5 and Figure 6.

2) REAL VEHICLE ATTACKS

In the second experiment, the WINDS is tested on the real-world vehicle attacks and compared with baseline

TABLE 7. Comparison of the WINDS with existing methods using real vehicle attack data.

Attack Type	IDS	Accuracy	Sensitivity (Recall)	Precision
Gear Spoofing	WINDS	0.9883	0.9845	0.9958
	SAIDuCANT	0.8262	0.9702	0.8245
	GIDS	0.9620	0.9650	0.9810
	DCNN	0.9995	0.9989	0.9999
	Frequency-based	0.9273	0.8770	0.9886
RPM Spoofing	WINDS	0.9926	0.9890	0.9986
	SAIDuCANT	0.8033	0.9636	0.8010
	GIDS	0.9800	0.9900	0.9830
	DCNN	0.9997	0.9994	0.9999
	Frequency-based	0.9472	0.9211	0.9815
Fuzzy Attack	WINDS	0.8778	0.8339	0.9816
	SAIDuCANT	0.8782	0.9958	0.8639
	GIDS	0.9800	0.995	0.9730
	DCNN	0.9982	0.9965	0.9995
	Frequency-based	0.8170	0.7556	0.9599
DoS Attack	WINDS	0.9497	0.9415	0.9797
	SAIDuCANT	0.9808	1.0000	0.9771
	GIDS	0.9790	0.9960	0.9680
	DCNN	0.9997	0.9989	1.0000
	Frequency-based	0.8711	0.8316	0.9617

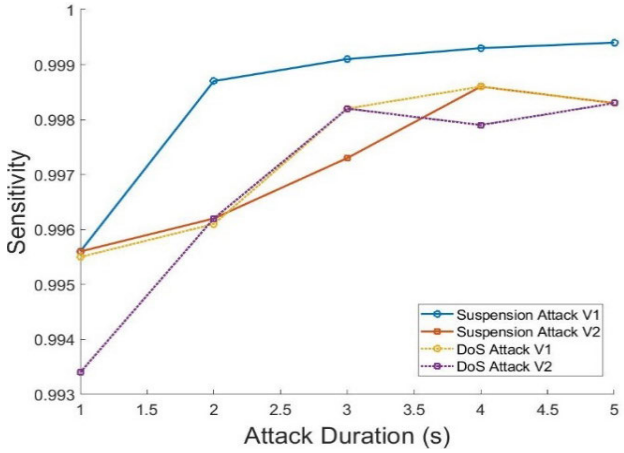

FIGURE 5. The sensitivity of WINDS algorithm during various suspension and DoS attacks. The sensitivity of the algorithm gets better with the rising attack duration.

frequency-based IDS and other existing methods; GIDS [11], DCNN [14], and SAIDuCANT [32], which are based on generative adversarial nets, deep convolutional neural network, and a specification of CAN timing, respectively. The results for the real-vehicle attacks are summarized in Table 7.

Gear spoofing and RPM attacks directly target certain IDs, and the WINDS can detect 98.45% and 98.90% of these attacks accordingly and provides over 99% precision for both cases.

The attack detection rate and accuracy decrease for Fuzzy and DoS attacks. This is partly because these attacks do not target any particular IDs; therefore, they are not as disruptive as direct attacks like gear or RPM spoofing. While the WINDS' sensitivity for the DoS attack is 94.15%, it can decrease to 83.39% for the fuzzy attack. This is an expected result; hence DoS attack was implemented using the highest ID number while the fuzzy attack transmits random ID numbers. Some of these IDs have low priority and have no disruption in transmitting authentic messages because of the arbitration process. We also compared WINDS with some alternative methods, including frequency-based IDS, which measures the frequency of attacked ID and generates an alarm if the threshold exceeds lower or higher threshold bonds. The WINDS outperforms the frequency-based IDS in all metrics for all attack types. Compared to GIDS and SAIDuCANT, the proposed method has advantages in generating better results for gear and RPM spoofing attack scenarios; however, its performance is slightly lower in the case of DoS and Fuzzy attacks. Although DCNN has the best performance for this dataset, it requires extensive training with attack data; because it involves a supervised learning method. It is also computationally expensive and requires GPU acceleration.

B. DISCUSSIONS

An IDS can be implemented as a host-based (also known as node-based) or network-based. In the host-based IDS, each ECU has an integrated IDS and may dismiss the message according to the IDS decision. However, this requires additional resources in each ECU. On the other hand, the network-based approach has only one IDS implemented on the gateway ECU. The WINDS is independent of implementation perspectives, suitable for implementation through host-based or network-based approaches with the same performance; however, the required resources would differ. This allows the method to be implemented on various applications, from low-end resource constraint vehicles as a network-based IDS to high-end vehicles as an advanced sensor for intrusion prevention systems in each ECU.

An IDS should satisfy specific requirements for vehicles, which are real-time safety-critical cyber-physical systems. In short, it should detect attacks correctly in an acceptable

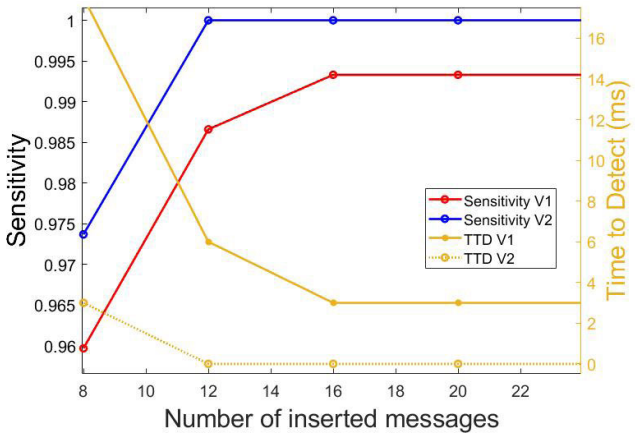

FIGURE 6. The sensitivity of WINDS algorithm during different replay attacks. The increased message insertion rate increases the sensitivity while decreasing the time to detect.

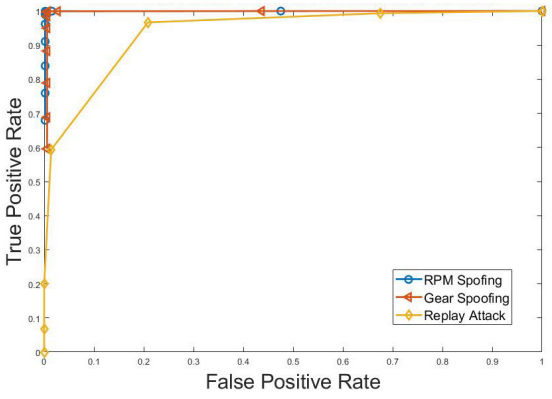

FIGURE 7. The Receiver Operating Characteristic (ROC) curves for varying threshold values for RPM spoofing, gear spoofing, and replay attack.

time frame while using limited resources and without causing false alarms. Therefore, WINDS is assessed based on three criteria: timing behavior, success rate, and resource usage.

1) TIME ANALYSIS

Successful IDS must detect attacks as soon as possible to prevent propagating misinformation and causing system misbehavior. A metric suitable for measuring the algorithm's behavior is TTD, which varies by the parameters like the sampling time and the threshold. Assessing WINDS by TTD demonstrated that an increase in the attack strength decreases the detection time, as presented in Figure 6.

The processing time is also an important parameter to evaluate timing behavior. Contrary to TTD, the processing time depends on the hardware and can be decreased by optimizing computational logic (e.g., CWT can be computed for only the last portion of the window and use the historical data for the rest). Actual processing time requires implementing the WINDS algorithm on an ECU, which is not covered in this research. As WINDS can be implemented as a network-based IDS, this can be ignored even on low-end vehicles using only one high-end automobile processor on the gateway ECU.

The WINDS algorithm can detect an anomaly in milliseconds. Considering the delay times in the CAN network [48], the algorithm should be suitable for the real-time analysis of most ECUs.

2) SUCCESS RATE

As the main parameter, changes in the message frequency should be observed by WINDS to detect attacks. The method cannot locate, for instance, impersonate attacks, where a node is suspended, and a malicious node transmits on behalf of the suspended one by causing the protocol error. However, this can be easily detected by counting the error frames. In contrast, the proposed algorithm successfully detects time variations, which enable WINDS to locate all the flooding attacks by analyzing only a single ID even though the attacker targets different IDs.

The threshold is the most critical parameter that affects the success rate. The lower threshold value will increase the detection rate, but it will also raise false alarms. Additionally, the threshold can be adjusted based on IDs and adapted to the arbitration process of the CAN for increasing the overall performance. This adaption will decrease the false alarms because lower priority IDs are not as punctual as the higher priority IDs due to the arbitration mechanism in CAN. The Receiver Operating Characteristic (ROC) curves in Figure 7 depict WINDS' behavior for three different attack models: replay attack, gear and RPM spoofing attacks. The result shows that WINDS provides a good performance characteristic.

An alternative way to increase the system's performance comes from the driving mode; hence, some ECUs are linked to different driving modes. Theoretically, the wavelet can detect this change and may give a false alarm during the transition. After a window passes the transition period, it does not provide a warning. This requires further investigation and testing on data from different driving modes.

3) RESOURCE USAGE

Vehicles are resource-constrained cyber-physical systems. Distributed ECUs have limited memory, computational power, and bandwidth. Therefore, optimum IDS should have low resource usage. The WINDS does not transmit any messages, so it does not have any effects on the bandwidth.

The memory usage of WINDS is directly proportional to the window size. It analyses the timing of the messages and does not need to store data bits. It only requires a single bit of memory storage as a flag identifying the message exists in the given sample time. Therefore, each ID requires n-bit memory equal to the window size, which is 16 bytes in this experiment. This is a very reasonable amount, even for low-end ECUs.

The CWT mainly drains computational power. If the scale is increased, the required power will increase, too. Efficient CWT algorithms are essential for making the IDS affordable for all ECUs. A way to reduce the algorithm's computational cost is to sacrifice some memories when ECU has limited computational power available. For instance, the n-bit window is not necessarily required to be transformed to wavelet coefficients as a whole each time. Instead, updating only some bits from the previous transform is sufficient while

keeping the rest unchanged. A partial updating of the last window results in the new window, which was expected to be transformed.

VI. FUTURE DIRECTIONS

Although the research results demonstrated so far in this paper through various tests, analysis, and evaluation are promising, further improvement is achievable by analyzing each wavelet scale individually. This additional improvement would be at the cost of higher complexity and computational needs. It is worth investigating alternative wavelet-based IDS systems such as Discrete Wavelet Transforms (DWT) and Maximal Overlap Discrete Wavelet Transform (MODWT); mainly, for reducing the computational cost, and conduct further assessments and comparisons with other techniques.

WINDS is limited to analyzing system behavior based on message frequency, and it is not extended toward nodes transmitting infrequent messages. The current implementation is not detecting attacks that do not affect the message frequency, requiring further investigations.

There is still a need for experimenting with real cars, considering various attack scenarios followed with suitable data collection to generate comprehensive and efficient datasets. Existing datasets available from open-access research centers are limited to specific cases, and yet, they don't provide essential system specifications under test and technical details of testing scenarios. This paper successfully demonstrated methods for generating synthetic attacks to overcome weaknesses from open-access datasets. This is limited to simple cases and leaves generation complex synthetic attacks, which are needed for sophisticated attacking scenarios, for future research. Furthermore, it is also crucial to test IDS on various driver and journey types, meaning that more datasets are needed for achieving efficient analysis.

The lack of available datasets for various vehicle models also prevents us from implementing an optimization process for the parameter decision. Optimization on limited datasets will cause overtraining. Therefore, it will worth investigating optimization techniques when we have enough independent datasets to improve the performance of WINDS.

For safe driving, it is essential to prevent attacks that cause system misbehavior. The current implementation of WINDS is designed as an intrusion detection system. To implement a real-time intrusion prevention system, each ID should be analyzed separately to gather its deadline. Then WINDS should be adapted to respond to the deadline. The prevention mechanisms to invalidate messages also need to be assessed and combined with WINDS. It is also worth mentioning that the WINDS is implemented on a personal computer. Although the TTD will be the same, the processing time will vary. Therefore, we aim to apply the WINDS on an ECU and gather processing time.

VII. CONCLUSION

In this paper, a wavelet-based intrusion detection mechanism is proposed to protect the in-vehicle communication network. It was shown that the WINDS algorithm could detect intrusions without any disruption to the network.

The proposed method can be applied to various vehicle models without any modification on the parameters. It has been tested on two datasets, which include replay, suspension, DoS, fuzzy, and spoofing attacks, collected from three vehicles. The results are also compared with other existing solutions.

Overall experimental results show that WINDS has a reasonable detection rate, in a short time from when the attack starts; therefore, it may be a suitable IDS for in-vehicle CAN networks.

REFERENCES

[1] T. Kosch, C. Schroth, M. Strassberger, and M. Bechler, *Automotive Internetworking*. Chichester, U.K.: Wiley, 2012.

[2] K. Matheus and T. Königseder, *Automotive Ethernet*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[3] B. Groza and P.-S. Murvay, "Security solutions for the controller area network: Bringing authentication to in-vehicle networks," *IEEE Veh. Technol. Mag.*, vol. 13, no. 1, pp. 40–47, Mar. 2018.

[4] T. Hoppe and J. Dittman, "Sniffing/replay attacks on CAN buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy," in *Proc. 2nd Workshop Embedded Syst. Secur. (WESS)*, 2007, pp. 1–6.

[5] S. Fröschle and A. Stühring, "Analyzing the capabilities of the CAN attacker," in *Proc. Eur. Symp. Res. Comput. Secur.*, in Lecture Notes in Computer Science, vol. 10492, Sep. 2017, pp. 464–482, doi: 10.1007/978-3-319-66402-6_27.

[6] C. Miller and C. Valasek. (2014). *A Survey of Remote Automotive Attack Surfaces*. Accessed: Mar. 31, 2018. [Online]. Available: https://www.ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf

[7] P.-S. Murvay and B. Groza, "DoS attacks on controller area networks by fault injections from the software layer," in *Proc. 12th Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2017, pp. 1–10, doi: 10.1145/3098954.3103174.

[8] N. Nowdehi, A. Lautenbach, and T. Olovsson, "In-vehicle CAN message authentication: An evaluation based on industrial criteria," in *Proc. IEEE 86th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2017, pp. 1–7, doi: 10.1109/VTCFall.2017.8288327.

[9] S. Nie, L. Liu, and Y. Du, "Free-fall: Hacking tesla from wireless to CAN bus," in *Proc. BlackHat USA*, 2017, pp. 1–16. Accessed: Nov. 30, 2018. [Online]. Available: https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf

[10] C. Wang, Z. Zhao, L. Gong, L. Zhu, Z. Liu, and X. Cheng, "A distributed anomaly detection system for in-vehicle network using HTM," *IEEE Access*, vol. 6, pp. 9091–9098, 2018, doi: 10.1109/ACCESS.2018.2799210.

[11] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in *Proc. 16th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2018, pp. 1–6, doi: 10.1109/PST.2018.8514157.

[12] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2016, pp. 130–139, doi: 10.1109/DSAA.2016.20.

[13] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS ONE*, vol. 11, no. 6, Jun. 2016, Art. no. e0155781, doi: 10.1371/journal.pone.0155781.

[14] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Veh. Commun.*, vol. 21, Jan. 2020, Art. no. 100198, doi: 10.1016/j.vehcom.2019.100198.

[15] M. Muter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 1110–1115, doi: 10.1109/IVS.2011.5940552.

[16] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *Proc. IEEE 2nd Int. Forum Res. Technol. Soc. Ind. Leveraging Better Tomorrow (RTSI)*, Sep. 2016, pp. 1–6, doi: 10.1109/RTSI.2016.7740627.

[17] W. Wu, Y. Huang, R. Kurachi, G. Zeng, G. Xie, R. Li, and K. Li, "Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks," *IEEE Access*, vol. 6, pp. 45233–45245, 2018, doi: 10.1109/ACCESS.2018.2865169.

[18] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 220–225, doi: 10.1109/IVS.2008.4621263.

[19] I. Studnia, E. Alata, V. Nicomette, M. Kaâniche, and Y. Laarouchi, "A language-based intrusion detection approach for automotive embedded networks," *Int. J. Embedded Syst.*, vol. 10, no. 1, pp. 1–12, 2018, doi: 10.1504/IJES.2018.089430.

[20] G. Dupont, J. Den Hartog, S. Etalle, and A. Lekidis, "Evaluation framework for network intrusion detection systems for in-vehicle CAN," in *Proc. IEEE Int. Conf. Connected Vehicles Expo (ICCVE)*, Nov. 2019, pp. 1–6, doi: 10.1109/ICCVE45908.2019.8965028.

[21] *Road Vehicles—Controller Area Network (CAN)—Part 1?: Data Link Layer and Physical Signalling*, Standard ISO 11898-1:2015, Technical Corrigendum, 2015, pp. 1–6.

[22] D. B. Percival and A. T. Walden, *Wavelet Methods for Time Series Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[23] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, "VoltageIDS: Low-level communication characteristics for automotive intrusion detection system," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 2114–2129, Aug. 2018.

[24] M. Spicer, A. L. Wicks, A. L. Abbott, S. C. Southward, and M. Spicer, "Intrusion detection system for electronic communication buses: A new approach," Virginia Polytech. Inst. State Univ., Blacksburg, VA, USA, Tech. Rep. 81863, 2017.

[25] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *Proc. 25th USENIX Secur. Symp.*, 2016, pp. 911–927. Accessed: May 31, 2018. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cho

[26] M. Muter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks," in *Proc. 6th Int. Conf. Inf. Assurance Secur.*, Aug. 2010, pp. 92–98, doi: 10.1109/ISIAS.2010.5604050.

[27] C. Young, H. Olufowobi, G. Bloom, and J. Zambreno, "Automotive intrusion detection based on constant CAN message frequencies across vehicle driving modes," in *Proc. ACM Workshop Automot. Cybersecurity, Co-Located With CODASPY (AutoSec)*, 2019, pp. 9–14, doi: 10.1145/3309171.3309179.

[28] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2016, pp. 63–68, doi: 10.1109/ICOIN.2016.7427089.

[29] S. Otsuka, T. Ishigooka, Y. Oishi, and K. Sasazawa, "CAN security: Cost-effective intrusion detection for real-time control systems," SAE Tech. Papers 2014-01-0340, Nov. 2014, doi: 10.4271/pt-174.

[30] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in *Proc. World Congr. Ind. Control Syst. Secur. (WCICSS)*, 2015, pp. 45–49, doi: 10.1109/WCICSS.2015.7420322.

[31] Y. Hamada, M. Inoue, H. Ueda, Y. Miyashita, and Y. Hata, "Anomaly-based intrusion detection using the density estimation of reception cycle periods for in-vehicle networks," *SAE Int. J. Transp. Cybersecur. Privacy*, vol. 1, no. 1, pp. 39–56, May 2018, doi: 10.4271/11-01-01-0003.

[32] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "SAIDuCANT: Specification-based automotive intrusion detection using controller area network (CAN) timing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1484–1494, Feb. 2020, doi: 10.1109/tvt.2019.2961344.

[33] H. Lee, S. H. Jeong, and H. K. Kim. (2017). *OTIDS?: A Novel Intrusion Detection System for In-Vehicle Network by Using Remote Frame*. [Online]. Available: https://www.ucalgary.ca/pst2017/files/pst2017/paper-67.pdf%0Ahttp://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset

[34] B. Groza and P.-S. Murvay, "Efficient intrusion detection with Bloom filtering in controller area networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 4, pp. 1037–1051, Apr. 2019, doi: 10.1109/TIFS.2018.2869351.

[35] A. Tomlinson, J. Bryans, S. A. Shaikh, and H. K. Kalutarage, "Detection of automotive CAN cyber-attacks by identifying packet timing anomalies in time windows," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2018, pp. 231–238, doi: 10.1109/DSN-W.2018.00069.

[36] M. Hamdi and N. Boudriga, "Detecting denial-of-service attacks using the wavelet transform," *Comput. Commun.*, vol. 30, no. 16, pp. 3203–3213, Nov. 2007, doi: 10.1016/j.comcom.2007.05.061.

[37] S.-Y. Ji, B.-K. Jeong, C. Kamhoua, N. Leslie, and D. H. Jeong, "Estimating attack risk of network activities in temporal domain: A wavelet transform approach," in *Proc. 11th IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2020, pp. 0826–0832, doi: 10.1109/uemcon51285.2020.9298153.

[38] P. Zuraniewski and D. Rincón. (2006). *Wavelet Transforms and Change-Point Detection Algorithms for Tracking Network Traffic Fractality*. Accessed: Jun. 25, 2019. [Online]. Available: https://ieeexplore.ieee.org/ielx5/11058/34932/01678244.pdf?tp=&arnumber=1678244&isnumber=34932&ref=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8=

[39] A. Palanca, E. Evenchick, F. Maggi, and S. Zanero, "A stealth, selective, link-layer denial-of-service attack against automotive networks," in *Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment*, in Lecture Notes in Computer Science, vol. 10327, Jul. 2017, pp. 185–206, doi: 10.1007/978-3-319-60876-1_9.

[40] H. Lee, K. Choi, K. Chung, J. Kim, and K. Yim, "Fuzzing CAN packets into automobiles," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2015, pp. 817–821, doi: 10.1109/AINA.2015.274.

[41] T. Hoppe, S. Kiltz, A. Lang, and J. Dittmann, "Exemplary automotive attack scenarios: Trojan horses for electronic throttle control system (ETC) and replay attacks on the power window system," in *Proc. Automat. Secur. VDI-Berichte Nr. VDI/VW Gemeinschaftstagung Automat. Secur.*, 2007, pp. 165–183. Accessed: Jun. 21, 2019. [Online]. Available: https://pdfs.semanticscholar.org/d10a/558b9caa8bd41c0112434f3b19eb8aab2b5c.pdf

[42] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, Sep. 1994, doi: 10.1093/biomet/81.3.425.

[43] B. A. Mozzaquatro, R. P. De Azevedo, R. C. Nunes, A. D. J. Kozakevicius, C. Cappo, and C. Schaerer, "Anomaly-based techniques for Web attacks detection," *J. Appl. Comput. Res.*, vol. 1, no. 2, pp. 111–120, Feb. 2012, doi: 10.4013/jacr.2011.12.06.

[44] G. Dupont, A. Lekidis, J. Den Hartog, and S. Etalle. (2019). Automotive controller area network (CAN) bus intrusion dataset v2. 4TU.ResearchData. Accessed: Dec. 9, 2019. [Online]. Available: https://data.4tu.nl/repository/uuid:b74b4928-c377-4585-9432-2004dfa20a5d

[45] H. K. Kim. Car-hacking dataset. Hacking Countermeasure Research Lab. Accessed: Dec. 1, 2020. [Online]. Available: https://sites.google.com/a/hksecurity.net/ocslab/Datasets/CAN-intrusion-dataset

[46] G. Peyr, "Mathematical foundations of data sciences," Tech. Rep., 2018. [Online]. Available: https://mathematical-tours.github.io/book/

[47] Q. Lin, S. Verwer, R. Kooij, and A. Mathur, "Using datasets from industrial control systems for cyber security research and education," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Secur.*, Oct. 2019, pp. 122–133, doi: 10.1007/978-3-030-37670-3_10.

[48] U. Klehmet, T. Herpel, K. S. Hielscher, and R. German, "Delay bounds for CAN communication in automotive applications," in *Proc. 14th GI/ITG Conf. Measuring, Modelling Eval. Comput. Commun. Syst.*, 2008, pp. 1–15.

MEHMET BOZDAL received the B.Sc. degree in electrical and electronics engineering from Çukurova University, Adana, Turkey, in 2013, and the M.S. degree in embedded systems engineering from the University of Leeds, Leeds, U.K., in 2016. He is currently pursuing the Ph.D. degree in security of in-vehicle communication with the Integrated Vehicle Health Management (IVHM) Centre, School of Aerospace, Transport, and Manufacturing (SATM), Cranfield University, U.K. His research interests include in-vehicle communication, embedded systems, cyber-physical systems, intrusion detection, and secure communication. He received the Ph.D. Scholarship from the Turkish Ministry of Education.

MOHAMMAD SAMIE received the B.Sc. degree in electronics from the Islamic Azad University of Saveh, Iran, in 1997, the M.Sc. degree in electronics from Shiraz University, Shiraz, Iran, in 2002, and the Ph.D. degree in advanced electronics from the University of the West of England, Bristol, U.K., in 2012. He is currently working as a Lecturer with the School of Aerospace, Transport, and Manufacturing (SATM), Cranfield University, U.K., where he is also leading Seretonix, a Secure and Reliable Electronic Systems Group, focusing on the resilience and security of electronics. He has accumulated a wide and varied experience in field-programmable gate arrays (FPGAs) and ASIC design, simulation, verification, and implementation (Toumaz in Didcot, U.K.). He was involved with two EPSRC-funded projects NFF and SABRE, where he was responsible for creating most of the detailed designs and implementations. He has published 43 international journals, conference papers, and book chapters, with two awards as best articles on bio-inspired electronics.

IAN K. JENNIONS received the degree in mechanical engineering and the Ph.D. degree in CFD from Imperial College London. He has worked for Rolls-Royce, General Electric, and Alstom in a number of technical roles, gaining experience in aerodynamics, heat transfer, fluid systems, and mechanical design. In July 2008, he moved to Cranfield University, as a Professor and the Director of the IVHM Centre, which is funded by a number of industrial companies, including Boeing, BAE Systems, Thales, Meggitt, MOD, DRS, Alstom Transport, and Novartis. He has led the development and growth of the IVHM Centre, in research and education, since its inception. His career spans over 40 years, working mostly for a variety of gas turbine companies. He has coauthored the book *No Fault Found: The Search for the Root Cause*. He is a Fellow of IMechE, RAeS, ASME, and PHM, a Contributing Member of the HM-1 IVHM Committee, the Chair of the SAE IVHM Steering Group, represents the Editorial Board of the *International Journal of Condition Monitoring*, the Director of the PHM Society, and a Chartered Engineer. He is an Editor of five SAE books on IVHM and the recent *The World of Civil Aerospace*.

2021-04-13

WINDS: A Wavelet-based Intrusion Detection System for Controller Area Network (CAN)

Bozdal, Mehmet

IEEE

Bozdal M, Samie M, Jennions IK. (2021) WINDS: A Wavelet-based Intrusion Detection System for Controller Area Network (CAN). IEEE Access, Volume 9, 2021, pp. 58621-58633
https://doi.org/10.1109/ACCESS.2021.3073057